DiMPLe - Disentangled Multi-Modal Prompt Learning: Enhancing Out-Of-Distribution Alignment with Invariant and Spurious Feature Separation

Supplementary Material

This section contains supplementary material that provides additional details for the main paper and further experimental analysis. This section follows the contents in the following order.

- Methodology - Preliminary Details.
- Early disentanglement.

A. Methodology - Preliminary Details

Contrastive Language-Image Pre-training (CLIP): In the paradigm of multi-modal training, CLIP [21] introduces a dual-encoder architecture for the vision and text modalities. Extensive pair of images and its associated caption is used to train these encoders. The caption encoded by the textual encoder f is mapped to a textual feature space whereas its associated image encoded by the vision encoder g is mapped on to a visual feature space. The ultimate goal is to align the image and text features belonging to a corresponding pair of images whereas repel other textual and visual features away through contrastive learning approach. For the downstream task of image classification, CLIP can be employed in a prompt-based approach. In zero-shot classification, rather than undergoing additional training on the specific target classes (e.g., ”dog” or ”car”) within the dataset, CLIP leverages hand-crafted text prompts such as “`a photo of a [CLASS]`”, pass it through the textual encoder and obtain the textual embeddings corresponding to c classes where $c \in \{1, ..., C\}$. Given a new image x_{test}, CLIP computes its prediction probability $p(c|x_{test})$ using the similarity between the image embeddings and embeddings of each class prompt, effectively identifying which class label best aligns with the image based on pre-existing learned associations. This process enables CLIP to perform classification without direct exposure to task-specific data, facilitating adaptability across various classification tasks.

$$p(c|x_{test}) = \frac{\exp(\mathtt{sim}(\mathbf{z}_{test}, \mathbf{w}_c)/\tau)}{\sum_{j=1}^{C} \exp(\mathtt{sim}(\mathbf{z}_{test}, \mathbf{w}_j/\tau)} \tag{11}$$

Context-Optimization for Language modality (CoOp): Owing to the tedious task of manual prompt-engineering to achieve optimal performance using the CLIP model, authors in [31] introduced a sophisticated methodology of using soft prompts instead of the hand-crafted ones to describe an image. These soft prompts are learnable textual vector denoted as $\mathbf{v} = [v_1, v_2, ..., v_L]$. Each $v_l, l \in 1, ..., L$ is a vector of the same dimension as the word embeddings and L is a hyper-parameter which controls the number of context tokens. Hence, CoOp introduces a soft (learnable) prompt represented as:

$$\mathbf{t} = [v_1, v_2, ..., v_L, \mathtt{CLASS}] \tag{12}$$

For the downstream task of image classification, CoOp adapts CLIP for few-shot transfer by fine-tuning the above continuous set of prompt vectors within its language component. For a given set of N training images $\mathcal{D}_{train} = (x_i, y_i)_{i=1}^{N}$, CoOp computes the class probabilities and minimizes the cross-entropy loss $\mathcal{L}_{ce}$ to tune the learnable prompt vector. This optimization problem is expressed as:

$$t^* = \arg\min_t \mathcal{L}_{\text{ce}}, \tag{13}$$

$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{N} y_i \log p(y_i|x_i), \tag{14}$$

$$p(y_i|x_i) = \frac{\exp(\mathtt{sim}(\mathbf{z}_i, \tilde{\mathbf{w}}_{y_i})/\tau)}{\sum_{j=1}^{C} \exp(\mathtt{sim}(\mathbf{z}_i, \tilde{\mathbf{w}}_j)/\tau)}, \tag{15}$$

with $\mathbf{z}_i = f(x_i)$ are the image embeddings for image x_i and $\tilde{\mathbf{w}}_{y_i} = g(\mathbf{t}_{y_i}) = g([\mathbf{v}, y_i])$ are the learned textual embeddings. Here $\mathcal{L}_{ce}$ is computed using the similarity between image and text embeddings $\mathbf{z}_i$, $\tilde{\mathbf{w}}_{y_i}$ respectively.

Context-Optimization with Decoupled Image Features (CoOp-OOD): One of the shortcomings of the CoOp approach is its poor generalization capabilities when faced with novel classes. The reason being learning of spurious correlations during the fine-tuning stage. Moreover, using features extracted from the frozen encoders make CoOp vulnerable to bias and inaccurate learning. To address this, authors in [31] propose CoOp-OOD which decouples the image features into invariant and spurious components through two projection layers, ϕ and ψ. These layers separate the features without additional encoders. To ensure effective decoupling, they employ a structured causal model that describes the relationships between variables using conditional independence to minimize the mutual information between the invariant and spurious image features.

Multi-Modal Prompt Learning (MaPLe): The concept of introducing learnable prompt vectors instead of hard-coded ones was an important step in the direction of prompt optimization. However, the generalization performance of CoOp dropped significantly on novel classes and as a result the authors evolved their learning strategy by conditioning these soft prompts on image instances during the fine-tuning stage. They called this method conditional context optimization (Co-CoOp) [30]. Inspired by this work, authors in [15] designed a Multi-Modal Prompt Learning (MaPLe) framework, which brings in multi-modal learnable prompt vectors instead of using conditioned soft prompts for the text modality alone. The language branch of CLIP consists of b learnable tokens P_i concatenated with fixed tokens W_0 corresponding to the [CLASS] embedding which gets processed through J transformer layers of the textual encoder. Similarly, the vision branch also consists of b learnable tokens $\tilde{P}_i$ concatenated with the input image tokens E_i. These tokens are processed through the transformer layers of the image encoder up to depth J. The multi-modal prompts are propagated through the associated transformer layers of both textual and visual encoders respectively, in order to achieve better generalization capability. Furthermore, MaPLe employs "branch-aware" prompt coupling, where vision prompts $\tilde{\mathbf{P}}$ are generated as linear projections of language prompts $\mathbf{P}$ using a function $\mathcal{F}(\cdot)$. This coupling enables mutual gradient propagation and alignment between the visual and textual features, improving performance.

B. Early disentanglement

To disentangle the invariant and spurious features across vision-language modalities before the encoding stage, we incorporate the following strategy:

Feature Decomposition via Early Prompting: We introduce two distinct learnable prompts for each modality (vision and language): one for the invariant features and another for the spurious features. These prompts are designed to capture and decouple the task-relevant information (invariant) from the noise or spurious correlations. For each modality, two prompts are introduced: (1) Invariant Textual Prompt ($P_{x,u}$): Initialized with a template focused on invariant features, e.g., "`a photo capturing core and invariant features of a [class]`". (2) Spurious Textual Prompt ($P_{x,s}$): Initialized with a template designed to capture spurious features, e.g., "`a photo with features that keep changing of a [class]`". Each prompt is a learnable vector that will evolve through the training process.

Projection of Textual Prompts to Visual Prompts: Once the invariant and spurious prompts for the text modality are initialized, they are projected into the visual space to influence the corresponding visual representations. This is achieved through a linear projection layer for both prompts: (1) The invariant textual prompt vector $P_{x,u}$ is projected into the visual space via a learnable linear layer μ_u, resulting in an invariant visual prompt $\tilde{P}_{x,u}$:

$$\tilde{P}_{x,u} = \mu_u(P_{x,u}) \tag{16}$$

Similarly, the spurious textual prompt vector $P_{x,s}$ is projected into the visual space using another learnable linear layer η_s, resulting in a spurious visual prompt $\tilde{P}_{x,s}$:

$$\tilde{P}_{x,s} = \eta_s(P_{x,s}) \tag{17}$$

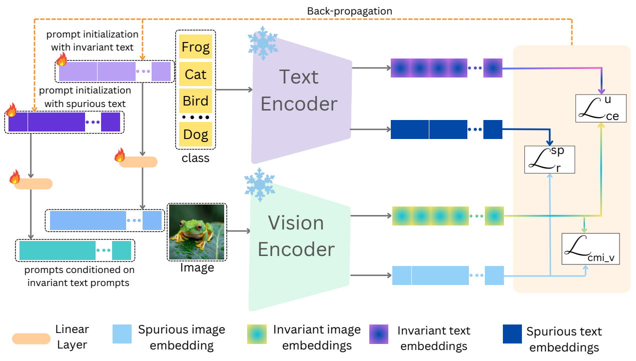

Figure 6. An overview of early disentanglement using separate prompt initializations to disentangle the features before encoding.

These projections allow for the disentanglement of the visual features in the shared visual-language space by guiding the vision branch's understanding of invariant and spurious components separately.

Disentangling Vision and Language Features: The disentanglement process happens early in the pipeline, before the cross-modal interaction, by ensuring that the vision prompts $\tilde{P}_{x,u}$ and $\tilde{P}_{x,s}$ guide the visual encoder in capturing invariant and spurious features, respectively. Similarly, the language encoder processes the invariant and spurious features separately through the respective prompt vectors $P_{x,u}$ and $P_{x,s}$. Let the text encoder output embeddings for the invariant and spurious prompts as $T_{x,u}$ and $T_{x,s}$, respectively, and the visual encoder outputs $V_{x,u}$ and $V_{x,s}$. The disentanglement occurs when these separate prompts influence the final embeddings produced by the vision and language encoders. The use of separate prompts for invariant and spurious features allows the model to disentangle the vision and language representations effectively. The vision encoder is guided to capture only the invariant features using the invariant prompts, while the spurious features are isolated using the spurious prompts. The same process applies to the language encoder, where the invariant and spurious prompts guide the encoding process at an early stage, ensuring that the invariant and spurious features are disentangled before the final multi-modal encoding.

Objective Function for Invariant & Spurious Features: Similar to the Late DiMPLe approach in order to ensure the learned representations align as intended, we introduce a triple-objective function:

(1) Invariant Alignment Loss ($\mathcal{L}^{u}_{ce}$): This loss minimizes the cross-entropy between the invariant visual features $V_{x,u}$ and the invariant textual embeddings $T_{x,u}$:

$$\mathcal{L}^{u}_{ce} = -\sum_{i=1}^{N} y_i \log p_u(y_i|V_{x,u}, T_{x,u}) \tag{18}$$

	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	*Average*
Base	66.43	97.57	95.10	66.07	92.10	87.20	29.07	74.23	78.33	76.70	76.93	76.34
Novel	62.80	92.93	95.27	69.80	68.70	89.80	27.43	69.73	50.90	70.23	69.63	69.75
HM	64.56	95.19	95.18	67.88	78.70	88.48	28.23	71.91	61.70	73.32	73.1	72.57

Table 7. Base-to-Novel Class Generalization using the early disentanglement method to separate the invariant and spurious features. The hyper-parameters were same as our main experiments.

(2) Spurious Unalignment Loss ($\mathcal{L}_r^{sp}$): A regularization term is applied to ensure the spurious features do not contribute to the classification. This is achieved by a uniformity constraint, ensuring the spurious components $V_{x,s}$ and $T_{x,s}$ do not align effectively with the target label:

$$\mathcal{L}_r^{sp} = \sum_{i=1}^{N} \ell_{KL}(p_s(y_i|V_{x,s}, T_{x,s})|p_0) \tag{19}$$

(3) Conditional Mutual Information Loss $\mathcal{L}_{cmi\_v}$: In Early DiMPLe, we enforce a conditional independence constraint between the invariant and spurious features for the vision modality alone. Specifically, we aim to minimize the conditional mutual information between the invariant and spurious components, conditioned on the label Y, using a mutual information regularization term $\mathcal{L}_{cmi_v}$. This is achieved through:

$$\mathcal{L}_{cmi\_v} = I(V_{x,i}, V_{x,s}|Y) \tag{20}$$

The total objective function for Early DiMPLe is then:

$$\mathcal{L} = \mathcal{L}_{ce}^{u} + \alpha \mathcal{L}_r^{sp} + \beta \mathcal{L}_{cmi\_v} \tag{21}$$

where α is a hyperparameter balancing the influence of the invariant and spurious loss terms.

The key difference in Early DiMPLe is that the image and text features are disentangled into invariant and spurious components at an early stage in the pipeline. As a result, the alignment loss is not calculated between the original image and text embeddings, but rather between their invariant components. This is why the loss function uses the invariant features $V_{x,u}$ and $T_{x,u}$ for the alignment, instead of the original image x_i and label y_i.

Base-to-Novel Class Generalization for early disentanglement. We evaluate the generalizability of the Early DiMPLe approach in a zero-shot setup, assessing its performance across both base and novel classes to highlight its capability to handle out-of-distribution (OOD) classes effectively. In this setting, each dataset is split into base classes (on which models are trained in a few-shot configuration) and novel classes, allowing us to measure how well the model can generalize to unseen categories without additional retraining. Table 7 provides a detailed analysis of Early DiMPLe's performance, focusing on its ability to separate invariant and spurious features before encoding, thereby enhancing generalization. From the results, it is evident that Early DiMPLe achieves high Base accuracy across datasets such as Caltech101 (97.57%) and OxfordPets (95.10%), demonstrating strong alignment with the base distribution. Similarly, the method shows competitive Novel accuracy, particularly on OxfordPets (95.27%) and Food101 (89.80%), highlighting its capacity to generalize effectively to unseen categories. The HM scores provide a more balanced perspective, with an average of 72.57% across datasets. Notable strengths are observed on datasets like Caltech101 (95.19%) and Flowers102 (78.70%), where the model maintains a favorable trade-off. However, challenges remain for datasets such as Aircraft and DTD, where lower HM values indicate difficulties in handling more complex or domain-specific novel distributions. This underscores potential areas for further refinement, such as incorporating more specialized backbones or advanced loss functions tailored to these datasets.

Computational complexity for Early vs late disentanglement: The early and late disentanglement in the DiMPLe framework reflects key differences in computational complexity. DiMPLe-E, with its early disentanglement approach, has a higher number of parameters and GFLOPs compared to DiMPLe, as mentioned in Table 8. Despite having a lesser number of trainable parameters and requiring comparatively lesser computations, DiMPLe's performance is better across all three evaluation settings: (1) Base-to-Novel Generalization, (2) Cross-dataset evaluation and (3) Domain Generalization.

Method	# Parameters	Parameters % CLIP	GFLOPs
DiMPLe	4.08M	3.28	1833.4
DiMPLe-E	6.72M	5.39	1903.8

Table 8. Computational complexity for late disentanglement (DiMPLe) vs early disentanglement (DiMPLe-E). We can see that DiMPLe is a better choice in terms of fewer GFLOPs and fewer tunable parameters compared to DiMPLe-E.